

# MODEL PENERAPAN ALGORITMA *ANT COLONY OPTIMIZATION* (ACO) UNTUK OPTIMASI SISTEM INFORMASI PENJADWALAN KULIAH

Rangga Sidik<sup>1</sup>, Mia Fitriawati<sup>2</sup>, Syahrul Mauluddin<sup>3</sup>, Agus Nursikuwagus<sup>4</sup>

<sup>1</sup>Program Studi Sistem Informasi, Universitas Komputer Indonesia, rangga.sidik@email.unikom.ac.id

<sup>2</sup>Program Studi Sistem Informasi, Universitas Komputer Indonesia, miafitriawati@email.unikom.ac.id

<sup>3</sup>Program Studi Manajemen Informatika, Universitas Komputer Indonesia, syahrul.mauluddin@email.unikom.ac.id

<sup>4</sup>Program Studi Manajemen Informatika, Universitas Komputer Indonesia, agus.nursikuwagus@email.unikom.ac.id

## ABSTRAK

Penjadwalan kuliah merupakan proses yang sangat kompleks, karena harus mengatur beberapa komponen seperti dosen, matakuliah, kelas, ruang dan waktu dengan memperhatikan sejumlah batasan dan syarat (*constraint*) tertentu. Dalam proses pembuatan jadwal kuliah ini setiap perguruan tinggi masing-masing berbeda syarat dan batasan. Hal ini lah yang menyebabkan banyak dilakukan penelitian terkait optimasi penjadwalan kuliah. Perlunya optimasi penjadwalan kuliah dirasakan juga oleh program studi manajemen yang merupakan salah satu program studi di Universitas Komputer Indonesia dengan jumlah mahasiswa yang banyak. Dalam proses penjadwalan kuliah masih dilakukan secara konvensional sehingga sering terjadi bentrok jadwal dan prosesnya membutuhkan waktu yang lama. Dalam penggambaran sistem ini menggunakan metode pendekatan berorientasi objek dengan alat bantu pemodelan UML (*Unified Modeling Language*) dan menggunakan metode pengembangan sistem model prototipe sebagai acuan tahapan penelitian. Penelitian ini merupakan tahap awal dalam membangun sistem informasi penjadwalan kuliah. Hasil penelitian ini berupa model penerapan algoritma *Ant Colony Optimization* (ACO) untuk optimasi sistem informasi penjadwalan kuliah.

## I. PENDAHULUAN

Di era digital ini segala pekerjaan ingin dikerjakan dengan serba cepat dan mudah. Hal ini tidak dapat kita pungkiri mengingat teknologi semakin berkembang dan segala kemudahan dapat diwujudkan. Khusus di instansi pendidikan seperti perguruan tinggi, salah satu proses yang memerlukan sentuhan teknologi adalah proses penjadwalan. Penjadwalan kuliah merupakan proses yang sangat kompleks, karena harus mengatur beberapa komponen seperti dosen, matakuliah, kelas, ruang dan waktu dengan memperhatikan sejumlah batasan dan syarat (*constraint*) tertentu. Dalam proses pembuatan jadwal kuliah ini setiap perguruan tinggi masing-masing berbeda syarat dan batasan dan pengerjaannya harus teliti serta harus selesai tepat waktu. Hal ini lah yang menyebabkan banyak dilakukan penelitian terkait optimasi penjadwalan kuliah.

Perlunya optimasi penjadwalan kuliah dirasakan juga oleh program studi manajemen yang merupakan salah satu program studi di Universitas Komputer Indonesia dengan jumlah mahasiswa yang cukup banyak dengan penerimaan mahasiswa setiap semester 200-250 mahasiswa. Dalam proses penjadwalan kuliah masih dilakukan secara konvensional sehingga sering terjadi bentrok jadwal dan membutuhkan waktu yang lama. Dengan adanya permasalahan tersebut kami melakukan penelitian awal yaitu membuat model sistem informasi penjadwalan kuliah dengan menerapkan algoritma *Ant Colony Optimization* (ACO) dengan tujuan untuk meminimalisir bentrok jadwal dan proses pembuatan jadwal yang lebih cepat.

## 2. TINJAUAN PUSTAKA

### 2.1. Definisi Penjadwalan Kuliah

Menurut Eddy Herjanto, “Penjadwalan adalah pengaturan waktu dari suatu kegiatan operasi, yang mencakup kegiatan mengalokasikan fasilitas, peralatan maupun tenaga kerja, dan menentukan urutan pelaksanaan bagi suatu kegiatan operasi”[1]. Adapun kuliah adalah “pelajaran yang diberikan di

perguruan tinggi”[2]. Berdasarkan definisi tersebut dapat diartikan penjadwalan kuliah adalah proses pengaturan waktu kuliah di sebuah perguruan tinggi.

## 2.2 Pengertian Optimasi

Menurut Suyanto [3] Istilah Optimasi bisa mengacu pada dua hal yang berbeda, yaitu mencari nilai minimal (minimasi) atau mencari nilai maksimal (maksimasi).

## 2.3 Algoritma ACO (*Ant Colony Optimization*)

“Algoritma semut adalah *bioinspired metaheuristic*, mempunyai sekelompok khusus yang berusaha menyalakan karakteristik kelakuan dari serangga sosial, yaitu koloni semut. Kelakuan dari tiap pelaku dalam meniru kelakuan dari semut hidup dan bagaimana mereka berinteraksi satu dengan lainnya agar dapat menemukan sumber makanan dan membawanya ke koloni mereka dengan efisien. Selama berjalan tiap semut mengeluarkan feromon, dimana semut lainnya sensitif dengan feromon tersebut sehingga memberikan harapan untuk mengikuti jejaknya. Lebih atau kurang intensitasnya tergantung pada konsentrasi dari feromon. Setelah beberapa waktu, jalur terpendek akan sering diikuti dan feromonnya menjadi jenuh [4].”

Tahapan Algoritma ACO (*Ant Colony Optimization*) [5] sebagai berikut :

Langkah 1: Inisialisasi harga parameter- parameter algoritma

Langkah 2: Pengisian titik pertama ke dalam tabu list

Langkah 3: Penyusunan rute kunjungan setiap semut ke setiap titik

Langkah 4: Perhitungan panjang rute setiap semut

Langkah 5: Perhitungan harga intensitas jejak kaki semut antar titik untuk siklus selanjutnya

Langkah 6: Pengosongan tabu list, dan ulangi langkah 2 jika diperlukan

Terkait penerapan algoritma semut pada penjadwalan kuliah telah banyak hasil penelitian dari para peneliti terdahulu. Berikut ini penelitian terdahulu yang sangat berkaitan dengan penelitian ini :

- 1) Selvia Lorena Br Ginting dan Hayi Akbar telah melakukan penelitian pembuatan perangkat lunak penjadwalan kuliah dengan menerapkan algoritma *Ant Colony Optimization* hasil penelitian menjelaskan bahwa Perangkat lunak penjadwalan kuliah dengan menerapkan metoda *Ant Colony* dapat melakukan pengaturan penjadwalan kuliah secara otomatis dan relatif cepat [5].
- 2) Komang Ayu Triana Indah, telah melakukan penelitian dengan judul ”Komparasi Metode *Ant Colony Optimization* dengan *Tabu Search* untuk Penjadwalan Perkuliahan”. Khusus terkait penerapan Algoritma *Ant colony Optimization*, penelitian ini menjelaskan bahwa aplikasi penjadwalan kuliah yang dibangun dapat mencapai optimal dengan tingkat kebenaran 90% [6].

## III. METODE PENELITIAN

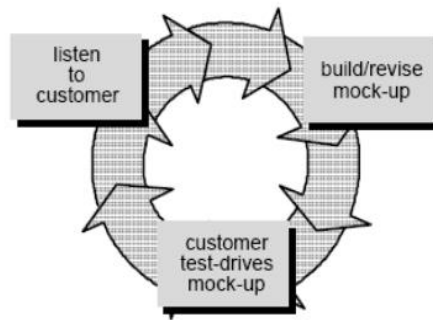
Dalam metode penelitian ini akan diuraikan meliputi metode pendekatan sistem, pengembangan sistem, lokasi penelitian dan teknik pengumpulan data.

### 3.1 Metode Pendekatan Sistem

Metode pendekatan sistem yang diterapkan dalam penelitian yaitu pendekatan berorientasi objek dengan *tools* pemodelan menggunakan UML.

### 3.2 Tahapan Penelitian

Tahapan-tahapan penelitian yang dilakukan berdasarkan metode pengembangan sistem model prototipe di bawah ini :



Gambar 1. Model Prototipe [7]

Berikut ini tahapan kegiatan dari model prototipe:

*Listen to customer.* Pada tahap ini analisis sistem menggali informasi terkait kebutuhan pengguna terhadap sistem informasi jadwal kuliah kemudian membuat model sistem informasi penjadwalan kuliah yang sedang berjalan.

*Build or revise Mock-up.* Ini adalah tahapan perancangan sistem informasi penjadwalan kuliah dengan menggunakan algoritma ACO (*Ant Colony Optimization*) yang meliputi perancangan database, perancangan antar muka serta koding dan uji coba pencarian nilai optimasi.

*Customer test-drive mock up.* Customer atau pengguna menguji sistem informasi penjadwalan kuliah sudah memenuhi kebutuhan atau belum. Apabila masih belum memenuhi kebutuhan maka mengulang ke tahap awal.

### 3.3 Lokasi Penelitian

Pada penelitian ini kami mengambil studi kasus di program Studi Manajemen, Universitas Komputer Indonesia yang beralamat di jalan dipatiukur No 102-118 Bandung.

### 3.4 Teknik Pengumpulan Data

Kegiatan pengumpulan data dilakukan dengan observasi di program studi manajemen. Kemudian untuk mendapatkan data yang diperlukan, kami mewawancarai sekretaris dan sekretariat Program Studi Manajemen. Teknik dokumentasi dilakukan untuk mendapatkan sumber data sekunder. Dokumen yang dikumpulkan seperti jadwal kuliah, berita acara perkuliahan, data ruang, data kelas, data dosen dan kurikulum.

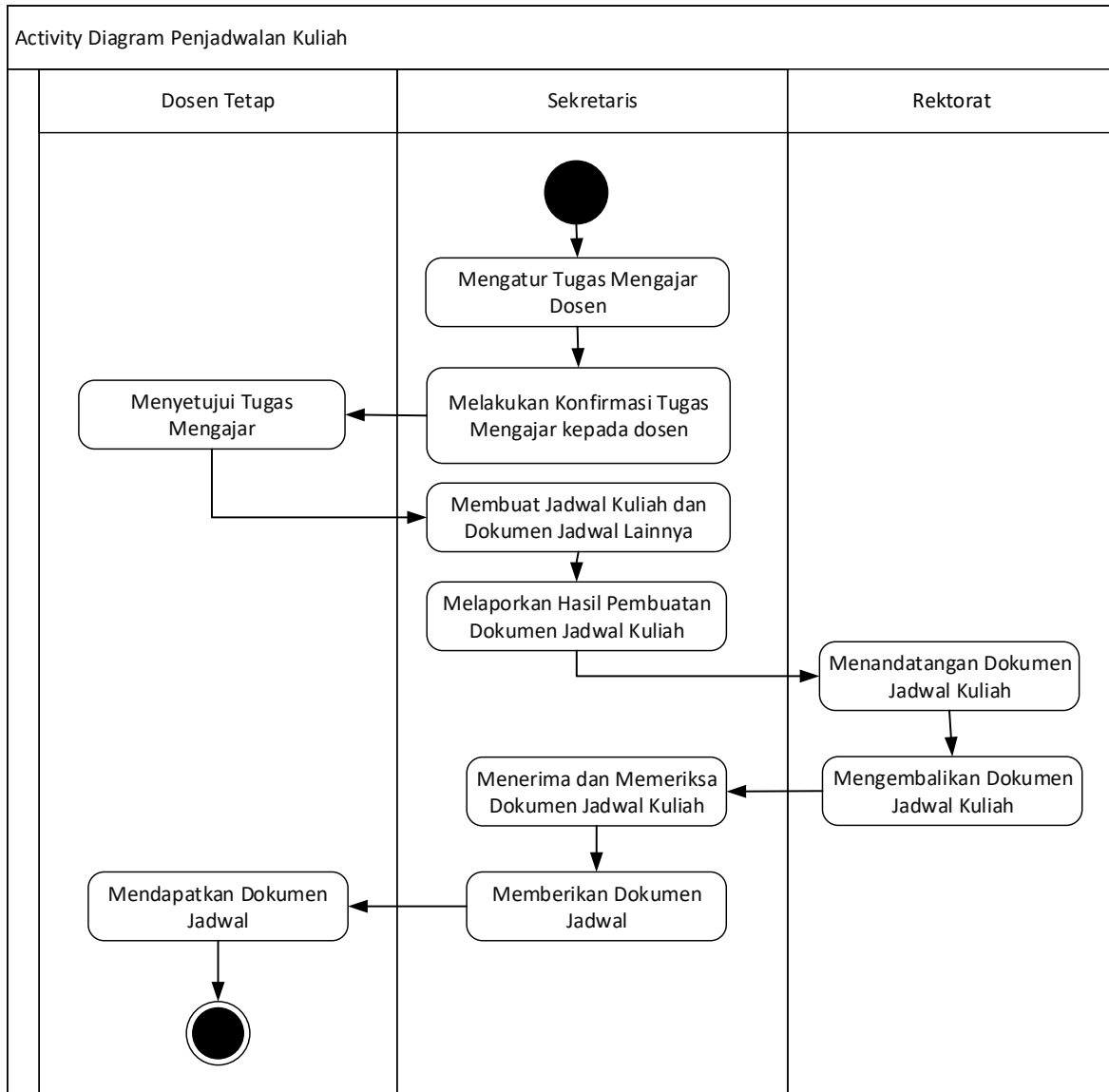
## IV. HASIL DAN PEMBAHASAN

Penelitian ini dilakukan dengan mengikuti langkah langkah pada metode pengembangan sistem model prototipe.

### 4.1 Listen to Customer

Pada tahap ini kami mewawancarai sekretaris dan sekretariat untuk mendapatkan hal yang dibutuhkan pengguna terkait sistem informasi penjadwalan kuliah kemudian membuat pemodelan pada sistem informasi penjadwalan kuliah yang sudah ada. Setelah melakukan wawancara, diketahui bahwa program studi ilmu manajemen membutuhkan sistem informasi penjadwalan kuliah untuk memudahkan pembuatan jadwal. Gambaran sistem yang ada sekarang dapat dilihat pada activity diagram gambar 2.

Hasil evaluasi pada sistem yang berjalan yaitu bahwa proses pembuatan jadwal masih dibuat secara konvensional dan diketahui bahwa proses penjadwalan kuliah memerlukan waktu yang lama dan jadwal yang dibuat sering/banyak mengalami bentrok sehingga sering mengalami perbaikan. Adapun solusi yang ditawarkan adalah membuat sistem informasi penjadwalan kuliah dengan algoritma ACO (*Ant Colony Optimization*) dengan harapan dapat menemukan proses penjadwalan yang paling optimal.



Gambar 2. Activity Diagram Penjadwalan Kuliah yang sedang berjalan

#### 4.2 Build or revise Mock-up

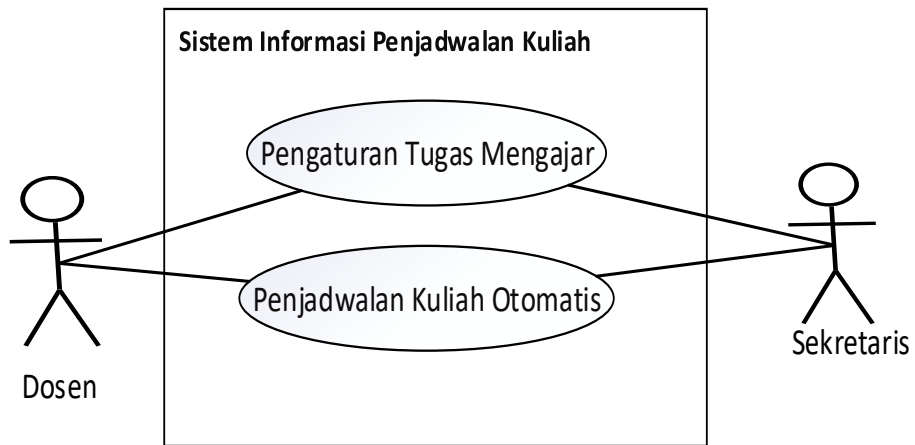
Hasil kegiatan *build or revise mock-up* yaitu perancangan fungsionalitas aplikasi, perancangan database dan perancangan antar muka.

##### 4.2.1 Rancangan Fungsionalitas Sistem

Fungsionalitas sistem akan digambarkan dengan *use case diagram* dan penjelasan kegiatan pada *use case diagram* akan dijelaskan pada *activity diagram*.

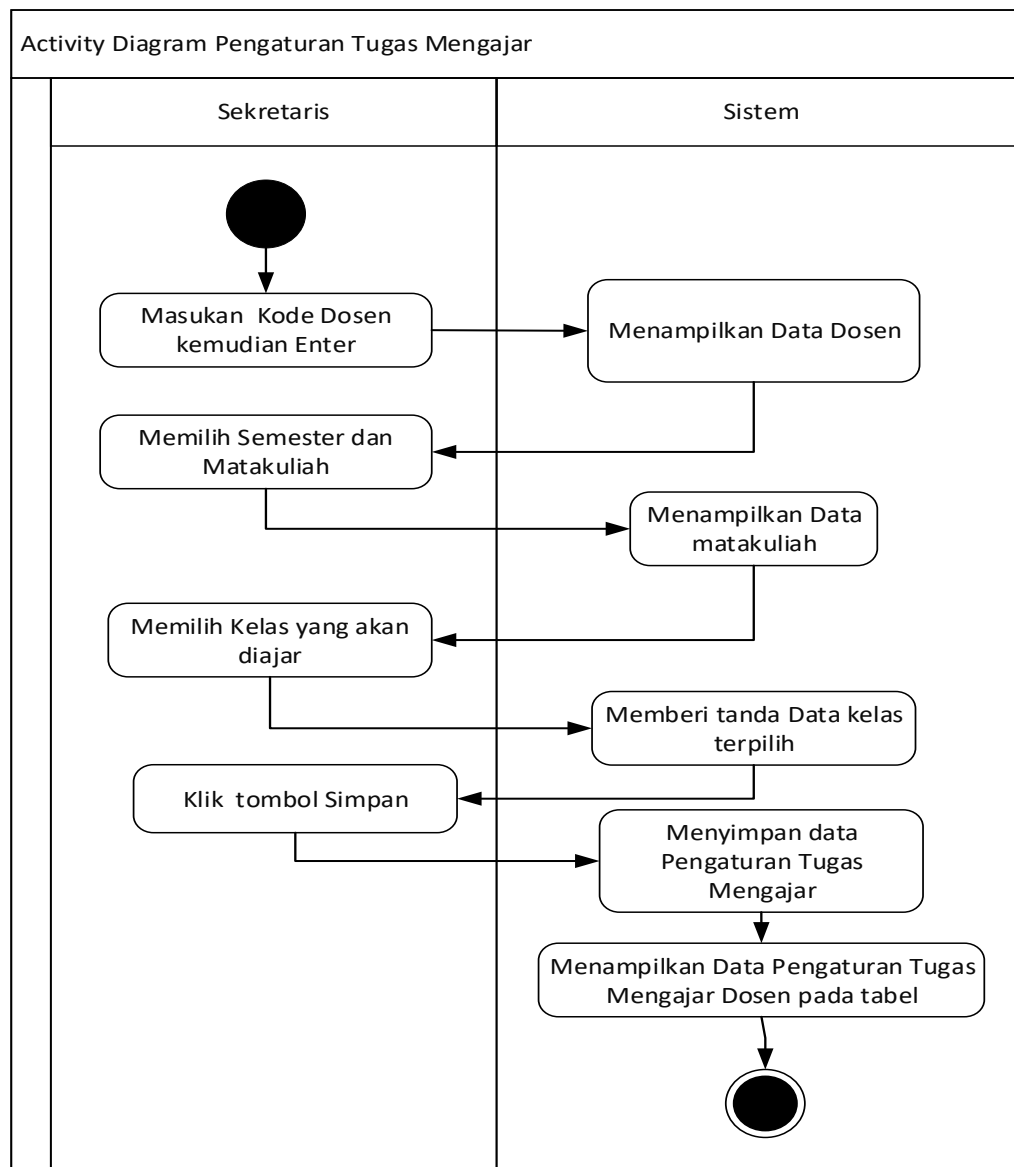
##### 1) Use Case Diagram

Use case Diagram sistem informasi penjadwalan kuliah terdiri dari dua aktor dan dua use case yang dapat dilihat pada gambar 3.

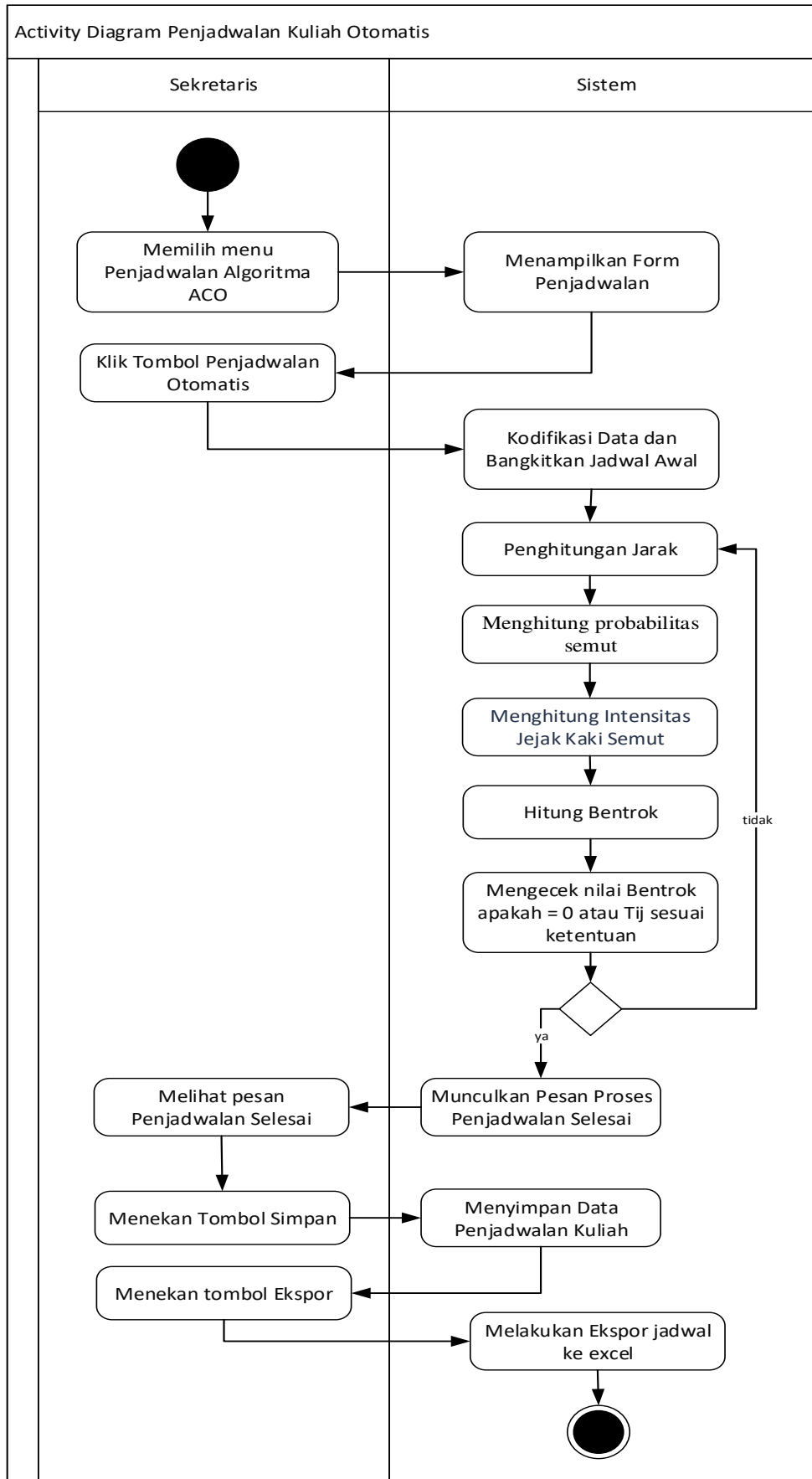


Gambar 3. Use Case Sistem informasi penjadwalan kuliah

## 2) Activity Diagram



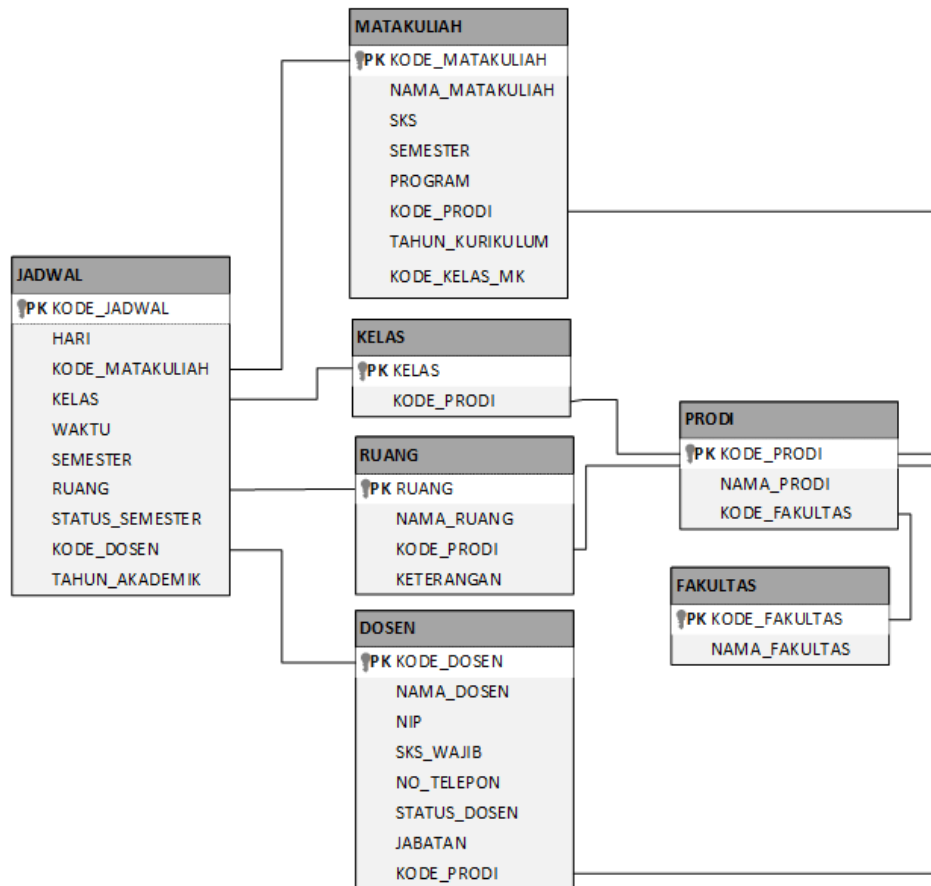
Gambar 4. Activity Diagram Pengaturan Tugas Mengajar



Gambar 5. Activity Diagram Penjadwalan Kuliah Otomatis

#### 4.2.2 Perancangan Database

Berikut ini rancangan database Sistem informasi penjadwalan kuliah:



Gambar 6. Rancangan Database Sistem informasi penjadwalan kuliah

#### 4.2.3 Perancangan Antar muka

Berikut ini hasil rancangan form penjadwalan kuliah otomatis:

Sistem Informasi Penjadwalan Kuliah Algoritma Semut

No	KODE DOSEN	NAMA DOSEN	KODE MATAKULIAH	MATAKULIAH	SKS	HARI
----	------------	------------	-----------------	------------	-----	------

PENJADWALAN OTOMATIS   SIMPAN DATA BASE   Export Excel

Gambar 7. Rancangan Form Penjadwalan Otomatis

#### 4.2.4 Pemodelan solusi dengan Algoritma ACO (*Ant Colony Optimization*)

Dalam penerapan algoritma ACO pada sistem informasi penjadwalan kuliah ini kami menerapkan beberapa asumsi sebagai berikut:

- 1) Semut adalah kombinasi dosen, mata kuliah dan kelas
- 2) Sumber makanan adalah pasangan ruang, hari dan jam
- 3) Feromon adalah jadwal yang diambil oleh dosen.
- 4) Jarak yang ditempuh adalah perbedaan nilai antara ruang, hari, dan jam.
- 5) Kegiatan pengaturan tugas mengajar atau pemasangan dosen, matakuliah dan kelas dilakukan oleh sekretaris, sehingga proses penjadwalan secara otomatis hanya untuk memasangkan hasil pengaturan tugas mengajar (dosen, matakuliah dan kelas) terhadap ruang, hari dan jam.
- 6) Penjadwalan kuliah ini untuk kasus kelas matakuliah, sehingga tidak menangani bentrok antar mahasiswa atau kelas itu sendiri.
- 7) Proses penjadwalan dibatasi untuk matakuliah teori di ruang kelas.
- 8) Jumlah hari yang disediakan sebanyak 6 hari dari senin sampai sabtu.
- 9) Setiap jadwal akan ditempatkan pada slot jam 3 sks.
- 10) Satu hari disediakan 4 slot jam dengan masing-masing 3 sks setiap jam.
- 11) Khusus hari jumat pada slot ke 3 tidak boleh dijadwalkan.
- 12) Khusus hari sabtu jumlah slot hanya 3.
- 13) Total slot jadwal (jumlah ruang x jumlah hari x jumlah slot jam) harus lebih banyak dari data yang akan dijadwalkan.
- 14) Batasan / *Constraints* terdiri dari:
  - a) Dosen tidak boleh memiliki 2 jadwal mengajar pada hari dan jam yang sama
  - b) Tidak boleh ada 2 jadwal pada ruang hari dan jam yang sama

Adapun langkah-langkah dalam penerapan algoritma semut dalam penjadwalan kuliah ini adalah sebagai berikut:

##### **Langkah 1a:** Inisialisasi Parameter

Pada kasus ini ada 5 parameter yang di berikan nilai awal yaitu:

1.  $\alpha$  (Alfa) = 0,01,  $\alpha$  adalah tetapan pengendali intensitas jejak semut, nilai  $\alpha \geq 0$ .
2.  $\beta$  (Betha) = 0,01,  $\beta$  adalah tetapan pengendali visibilitas, nilai  $\beta \geq 0$ .
3.  $\rho$  (Rho) = 0,03, nilai  $\rho$  harus  $> 0$  dan  $< 1$  untuk mencegah jejak pheromone yang tak terhingga,  $\rho$  merupakan tetapan penguapan jejak semut
4.  $\tau_{ij}$  Awal = 0,01  $\tau_{ij}$  adalah Intensitas jejak semut antar titik dan perubahannya
5.  $Q=1$ ,  $Q$  adalah tetapan siklus semut ( $Q$ )
6.  $NC_{max}$  tidak diberikan nilai awal karena jumlah siklus diketahui ketika penjadwalan menemui salah satu dari 2 syarat yaitu jika nilai bentrok mencapai 0 atau selisih nilai  $\tau_{ij}$  awal dan sebelumnya  $< 0,003$ ,  $NC_{max}$  adalah jumlah siklus maksimum.

##### **Langkah 1b:** Membangkitkan Semut/Jadwal Awal

Pada tahap ini dilakukan pengkodean untuk data-data yang akan diproses dengan algoritma ACO. Tahap membangkitkan semut/jadwal awal adalah tahap inisialisasi titik pertama setiap semut. Implementasinya adalah memasangkan secara acak data tugas mengajar (dosen, matakuliah, kelas)



dengan data ruang, hari dan jam. Masing-masing nilai ruang, hari dan jam didapatkan secara acak dengan fungsi random, sehingga terbentuk jadwal awal secara acak. Ilustrasi hasil pasangan tugas mengajar dengan ruang, hari dan jam seperti pada gambar 11.

	Dosen	MK	Kelas	Ruang	Hari	Jam
J1	1	1	5	2	4	1
J2	4	3	4	1	2	2
J3	3	2	3	2	4	1
J4	1	4	2	1	2	2
J5	2	5	1	3	2	2
J6	5	1	2	1	1	3

Gambar 11. Contoh tahap membangkitkan semut

**Langkah 2:** Pengisian tabu list dengan nilai titik pertama setiap semut [5]. Contoh pengisian titik pertama ke dalam tabu list seperti gambar 11.

**Langkah 3:** Penyusunan rute kunjungan setiap semut ke semua titik.

Pada tahap ini masing-masing jadwal/semut dipasangkan ke semua jadwal untuk menghitung jaraknya. Pengukuran jarak dihitung menggunakan rumus persamaan:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \dots\dots\dots (1)$$

Penerapan pada kasus ini dijelaskan perhitungan dij sebagai berikut:

$$d_{ij} = \sqrt{(\text{Jarak DHJ} + \text{Jarak RHJ})}$$

Jarak DHJ = Jarak Dosen<sup>2</sup> + Jarak Hari<sup>2</sup> + Jarak Jam<sup>2</sup>  
 Jarak RHJ = Jarak Ruang<sup>2</sup> + Jarak Hari<sup>2</sup> + Jarak Jam<sup>2</sup>  
 Jarak Dosen adalah nilai dosen ke i – nilai dosen ke j  
 Jarak Ruang adalah nilai ruang ke i – nilai ruang ke j  
 Jarak Hari adalah nilai hari ke i – nilai hari ke j  
 Jarak Jam adalah nilai jam ke i – nilai jam ke j

Berikut ini contoh perhitungan jarak untuk semua jadwal terhadap Jadwal J1. Jadwal J1 adalah sebagai berikut:

	D	R	H	J
J1	1,00	2,00	4,00	1,00

Maka untuk perhitungan jarak dari semua jadwal ke J1 dapat dilakukan seperti pada gambar 12.

Jadwal Ke	D	R	H	J	D J1	R J1	H J1	J J1	d DHJ	d RHJ	dij
J1	1,00	2,00	4,00	1,00	1,00	2,00	4,00	1,00	0,00	0,00	0,00
J2	4,00	1,00	2,00	2,00	0,00	3,00	-1,00	-2,00	14,00	6,00	4,47
J3	3,00	2,00	4,00	1,00	2,00	0,00	0,00	0,00	4,00	0,00	2,00
J4	1,00	1,00	2,00	2,00	0,00	-1,00	-2,00	1,00	5,00	6,00	3,32
J5	2,00	3,00	2,00	2,00	1,00	1,00	-2,00	1,00	6,00	6,00	3,46
J6	5,00	1,00	1,00	3,00	4,00	-1,00	-3,00	2,00	26,00	14,00	6,32

Gambar 12. Contoh penghitungan Jarak

Setelah melakukan proses seperti gambar 12 untuk masing-masing jadwal, berikut ini hasil penghitungan jarak (dij):

Colony	J1	J2	J3	J4	J5	J6
J1	0,00	4,47	2,00	3,32	3,46	6,32
J2	4,47	0,00	3,46	3,00	3,46	2,00
J3	2,00	3,46	0,00	3,87	3,46	5,29
J4	3,32	3,00	3,87	0,00	3,00	4,36
J5	3,46	3,46	3,46	3,00	0,00	4,47
J6	6,32	2,00	5,29	4,36	4,47	0,00

Gambar 13. Hasil perhitungan dij semua jadwal

Selanjutnya menghitung visibilitas antar titik ( $\eta_{ij}$ ).  $\eta$  digunakan dalam persamaan probabilitas titik yang akan dikunjungi. Nilai  $\eta$  merupakan hasil dari  $1/dij$  (jarak antar titik). Hasil perhitungan visibilitas antar titik ada pada gambar 14.

Colony	J1	J2	J3	J4	J5	J6
J1	0,00	0,22	0,50	0,00	0,29	0,16
J2	0,22	0,00	0,29	0,33	0,29	0,50
J3	0,00	0,29	0,00	0,00	0,29	0,19
J4	0,00	0,33	0,26	0,00	0,33	0,00
J5	0,29	0,00	0,29	0,33	0,00	0,22
J6	0,16	0,50	0,19	0,00	0,22	0,00

Gambar 14. Hasil perhitungan visibilitas antar titik

**Langkah 4:** Mencari probabilitas titik untuk dikunjungi.

Probabilitas dari Jadwal mulai J1 ke jadwal selanjutnya menggunakan rumus sebagai berikut:

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k' \in [N-tabu_k]} [\tau_{ik'}]^\alpha [\eta_{ik'}]^\beta} \dots\dots\dots (2)$$

Contoh hasil perhitungan probabilitas terhadap jadwal ke 1 atau J1:

Tabel 4. Probabilitas Jadwal terhadap J1

Jadwal	$\eta_{ij}$ J1	$\tau_{ij}$ awal	$\tau_{ij}$ awal <sup><math>\alpha</math></sup> * $\eta_{ij}$ <sup><math>\beta</math></sup>	Pij	Prob. Kumulatif
J1	0,00	0,01	0,00000000	0,00000	0,00000
J2	0,22	0,01	0,00223714	0,19105	0,19105
J3	0,50	0,01	0,00500000	0,42700	0,61805
J4	0,00	0,01	0,00000000	0,00000	0,61805
J5	0,29	0,01	0,00289017	0,24682	0,86487
J6	0,16	0,01	0,00158228	0,13513	1,00000
Jumlah			0,011709588		

Lakukan perhitungan yang sama untuk J2 sampai J6.

**Langkah 5:** Menentukan arah semut dengan melakukan random dan mencari probabilitas kumulatif yang dekat dengan nilai random

Mengacu ke tabel 4, misalkan nilai random yang dihasilkan adalah 0,76 maka arah semut adalah dari J1 ke J5 karena nilai 0,76 mendekati nilai probabilitas kumulatif 0,86487 milik J5. Kemudian dilakukan untuk jadwal yang lainnya dan dihasilkan arah seperti berikut ini:

Tabel 4. Arah semut

Semut	Rute	
1	J1	J5
2	J2	J4
3	J3	J5
4	J4	J2
5	J5	J3
6	J6	J4

**Langkah 6:** Perhitungan panjang rute setiap semut.

Perhitungan panjang rute tertutup (*length closed tour*) atau  $L_k$  setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut [5]. Secara umum  $L_k$  dihitung dengan persamaan berikut:

$$L_k = d_{tabu_k(n), tabu_k(1)} + \sum_{s=1}^{n-1} d_{tabu_k(s), tabu_k(s+1)} \dots\dots\dots (3)$$

dengna  $d_{ij}$  merupakan jarak antara titik  $i$  ke titik  $j$  yang dihitung dengan persamaan:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Pada kasus ini nilai  $L_k$  diambil dari nilai  $d_{ij}$  saja. Mengacu kepada gambar 13 hasil perhitungan  $d_{ij}$ , maka panjang rute masing-masing sebagai berikut:

Tabel 5. Panjang rute setiap semut

Semut	Rute		Lk
1	J1	J5	3,46
2	J2	J4	3,00
3	J3	J5	3,46
4	J4	J2	3,00
5	J5	J3	3,46
6	J6	J4	4,36

**Langkah 7:** Perhitungan perubahan harga intensitas jejak kaki semut antar titik

Rumus persamaan perubahan ini yaitu:

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \dots\dots\dots (4)$$

Dengan  $\Delta\tau_{ij}^k$  adalah perubahan harga intensitas jejak kaki semut antar titik setiap semut yang dihitung berdasarkan persamaan:

$$\Delta\tau_{ij}^k = \frac{Q}{L_k}, \text{ untuk } (ij) \in \text{titik asal dan titik tujuan dalam } tabu_k$$

$$\Delta\tau_{ij}^k = 0, \text{ untuk } (ij) \text{ lainnya}$$

Tabel 6. Perhitungan  $\Delta\tau_{ij}$ 

Q	Lk	Q / Lk
1	3,46	0,14451
1	3,00	0,16667
1	3,46	0,14451
1	3,00	0,16667
1	3,46	0,14451
1	4,36	0,11468
$\Delta\tau_{ij}$		0,88154

**Langkah 8:** Menentukan intensitas jejak semut antar titik ( $\tau_{ij}$ ) yang baru untuk siklus selanjutnya dengan rumus:

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta\tau_{ij} \quad \dots\dots\dots (5)$$

Berdasarkan nilai inisialisasi  $\rho$  bernilai 0,03,  $\tau_{ij}$  awal bernilai 0,01 maka:

$$\begin{aligned} \tau_{ij} \text{ Baru} &= (0,03 \times 0,01) + 0,88154; \\ &= 0,8818 \end{aligned}$$

**Langkah 9:** Menghitung Bentrok Jadwal

Pada aplikasi ini berhentinya proses penjadwalan akan ditentukan oleh 2 kondisi yaitu jika bentrok = 0 atau selisih nilai  $\tau_{ij}$  baru dan nilai  $\tau_{ij}$  sebelumnya mencapai nilai yang dipersyaratkan misalnya kurang dari 0,0003.

Untuk menghitung bentrok setiap jadwal dihitung dengan menjumlahkan bentrok DHJ (Dosen, Hari, Jam) dengan bentrok RHJ (Ruang, Hari, Jam).

$$\text{Bentrok} = \text{Bentrok DHJ} + \text{Bentrok RHJ}$$

Gambar 12 dan 13 di bawah ini menjelaskan contoh jadwal yang bentrok.

	Dosen	MK	Kelas	Ruang	Hari	Jam
J1	1	1	5	2	3	2
J2	2	2	4	1	5	3
J3	2	3	2	1	5	3
J4	6	5	1	3	4	3
J5	3	4	3	2	3	2
Jn	...	...	...	...	...	...

Gambar 12. Contoh Bentrok DHJ

	Dosen	MK	Kelas	Ruang	Hari	Jam
J1	1	1	5	2	3	2
J2	2	2	4	1	5	3
J3	2	3	2	1	5	3
J4	6	5	1	3	4	3
J5	3	4	3	2	3	2
Jn	...	...	...	...	...	...

Gambar 13. Contoh Bentrok RHJ

Semua bentrok kemudian dijumlahkan, contoh hasil rekapitulasi jumlah bentrok dapat dilihat pada tabel 6.

Tabel 6. Contoh rekapitulasi jumlah bentrok

No	Bentrok DHJ	Bentrok RHJ	Bentrok
J1	0	1	1
J2	1	1	2
J3	1	1	2
J4	0	0	0
J5	0	1	1
Jn	...	...	...

Mengacu pada tabel 6, Misalkan misalkan yang bentrok hanya pada J1 sampai J5 maka jumlah total bentrok = 6.

Langkah selanjutnya jika bentrok masih  $> 0$  atau selisih nilai  $\tau_{ij}$  baru dan nilai  $\tau_{ij}$  sebelumnya belum mencapai nilai yang dipersyaratkan, maka akan berlanjut ke langkah perbaikan. Pada tahap ini akan dilakukan proses pengacakan nilai ruang, hari dan jam untuk merubah nilai ruang, hari dan jam pada jadwal yang bentrok. Setelah melakukan perbaikan maka akan diproses ke tahap awal lagi.

## V. KESIMPULAN

Hasil penelitian ini berupa model sistem informasi penjadwalan kuliah menggunakan algoritma ACO. Adanya model ini diharapkan menjadi dasar membangun sistem informasi penjadwalan kuliah dalam meminimalisir bentrok jadwal dengan proses penjadwalan optimal.

## UCAPAN TERIMA KASIH

Kami mengucapkan terimakasih sebesar-besarnya kepada Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian Riset Teknologi dan Pendidikan Tinggi yang sudah memberikan dana hibah penelitian tahun anggaran 2018 untuk membiayai penelitian ini dengan no kontrak 023M/SP/LPPM/UNIKOM/III/2018

## DAFTAR PUSTAKA

- [1] E. Herjanto, *Manajemen Operasi*. Jakarta: Grasindo, 2007.
- [2] P. B. Kemdikbud, *Kamus Besar Bahasa Indonesia*. <https://kbbi.web.id/kuliah>, 2012.
- [3] Suyanto, *Evolutionary Computation Komputasi Berbasis Evolusi dan Genetika*. Bandung: Informatika, 2008.
- [4] H. Saragih, G. Hoendarto, B. Reza, dan D. Setiyadi, "Aplikasi Sistem Perangkat Lunak Menggunakan Algoritma Ant Untuk Mengatur Penjadwalan Kuliah," *Tek. dan Ilmu Komput.*, vol. 1, no. 3, hal. 241–256, 2012.
- [5] S. L. B. Ginting dan H. Akbar, "Pembangunan perangkat lunak menggunakan algoritma ant colony optimization untuk optimalisasi penjadwalan kuliah," *J. Manaj. Inform.*, vol. 01, no. 05, hal. 1–20, 2014.
- [6] A. K. I. Triana, "Komparasi Metode Ant Colony dengan Tabu Search untuk Penjadwalan Perkuliahan," *Matrix J. Manaj. Teknol. dan Inform.*, vol. 6, no. 3, hal. 148–156, Feb 2016.
- [7] H. Al Fatta, *Analisis Dan Perancangan Sistem Informasi Untuk Keunggulan Bersaing Perusahaan Dan Organisasi Modern*. Yogyakarta: Andi, 2007.